An Absolute 2-Approximation Algorithm for Two-Dimensional Bin Packing

Rolf Harren and Rob van Stee*

Max-Planck-Institut für Informatik (MPII),
Campus E1 4, D-66123 Saarbrücken, Germany.
{rharren,vanstee}@mpi-inf.mpg.de

Abstract. We consider the problem of packing rectangles into bins that are unit squares, where the goal is to minimize the number of bins used. All rectangles have to be packed non-overlapping and orthogonal, i.e., axis-parallel. We present an algorithm for this problem with an absolute worst-case ratio of 2, which is optimal provided $\mathcal{P} \neq \mathcal{NP}$.

Keywords: bin packing, rectangle packing, approximation algorithm, absolute worst-case ratio

1 Introduction

In the two-dimensional bin packing problem, a list $I = \{r_1, \ldots, r_n\}$ of rectangles of width $w_i \leq 1$ and height $h_i \leq 1$ is given. An unlimited supply of unit-sized bins is available to pack all items from I such that no two items overlap and all items are packed axis-parallel into the bins. The goal is to minimize the number of bins used. The problem has many applications, for instance in stock-cutting or scheduling on partitionable resources. In many applications, rotations are not allowed because of the pattern of the cloth or the grain of the wood. This is the case we consider in this paper.

Most of the previous work on rectangle packing has focused on the *asymptotic* approximation ratio, i.e., the long-term behavior of the algorithm. The asymptotic approximation ratio is defined as follows. Let $\textsc{alg}(I)$ be the number of bins used by algorithm $\textsc{alg}$ to pack input I. Denote the optimal number of bins by $\mathrm{OPT}(I)$. The asymptotic approximation ratio of a two-dimensional bin packing algorithm $\textsc{alg}$ is defined to be

$$\limsup_{n\to\infty} \sup_{I} \left\{ \frac{\textsc{alg}(I)}{\mathrm{OPT}(I)} \;\middle|\; \mathrm{OPT}(I) = n \right\}.$$

Caprara[5] was the first to present an algorithm with an asymptotic approximation ratio less than 2 for two-dimensional bin packing. Indeed, he considered 2-stage packing, in which the items must first be packed into shelves that are then packed into bins, and showed that the asymptotic worst case ratio between two-dimensional bin packing and 2-stage packing is $T_\infty = 1.691\ldots$. Therefore the asymptotic $\mathcal{FPTAS}$ for 2-stage packing from Caprara, Lodi & Monaci[6] achieves an asymptotic approximation guarantee arbitrary close to T_∞.

Recently, Bansal, Caprara & Sviridenko[2] presented a general framework to improve subset oblivious algorithms and obtained asymptotic approximation guarantees arbitrarily close to $1.525\ldots$ for packing with rotations of 90 degrees or without rotations. These are the currently best-known asymptotic approximation ratios for these problems. For packing squares into square bins, Bansal, Correa, Kenyon & Sviridenko[4] gave an asymptotic $\mathcal{PTAS}$.

* Research supported by German Research Foundation (DFG)

On the other hand, the same paper showed the $\mathcal{APX}$-hardness of two-dimensional bin packing without rotations, thus no asymptotic $\mathcal{PTAS}$ exists unless $\mathcal{P} = \mathcal{NP}$. Chlebík & Chlebíková[7] were the first to give explicit lower bounds of $1 + 1/3792$ and $1 + 1/2196$ on the asymptotic approximability of rectangle packing with and without rotations, respectively.

It should be noted that for some of the positive results mentioned above, the approximation ratio only gets close to the stated value for very large inputs. In particular, the 1.525-approximation from Bansal et al.[2] has an additive constant which is not made explicit in the paper but which the authors believe is extremely large[1]. Thus, for any reasonable input, the actual (absolute) approximation ratio of their algorithm is much larger than 1.525, and it therefore makes sense to consider alternative algorithms and in particular, an alternative performance measure.

In the current paper, we consider the absolute approximation ratio. This is defined simply as $\sup_I \text{ALG}(I)/\text{OPT}(I)$, where the supremum is taken over all possible inputs. Proving a bound on the absolute approximation gives us a performance guarantee for all inputs, not just for (very) large ones. Zhang[15] presented an approximation algorithm with an absolute approximation ratio of 3 for the problem without rotations. For the special case of packing squares, van Stee[14] showed that an absolute 2-approximation is possible. Additionally, Harren & van Stee[8] presented an absolute 2-approximation for packing rectangles in the case that 90 degree rotations are allowed. They also showed that the algorithm Hybrid First Fit has an absolute approximation ratio of 3 for packing without rotations, as conjectured by Zhang[15].

A related two-dimensional packing problem is the strip packing problem, where the items have to be packed into a strip of unit basis and unlimited height such that the height is minimized. Steinberg[13] and Schiermeyer[12] presented absolute 2-approximation algorithms for strip packing without rotations. We use Steinberg's algorithm in particular as a subroutine in our algorithm.

Our contribution. We present an approximation algorithm for two-dimensional bin packing with an absolute approximation ratio of 2. As Leung et al.[11] showed that it is strongly $\mathcal{NP}$-complete to decide wether a set of *squares* can be packed into a given square, this is best possible unless $\mathcal{P} = \mathcal{NP}$.

The two-dimensional bin packing problem can also be seen as a scheduling problem. Here the tasks $r_i = (w_i, h_i)$ have a running time w_i and need h_i consecutive machines to be proceeded. The restriction to a polynomial number of machines can be solved with methods from Jansen & Thöle[9], but the unrestricted version that we consider in this paper appears to be considerably more difficult.

2 Important tools and definitions

In this section we give the necessary definitions and introduce results that are important for our work.

Let $I = \{r_1, \ldots, r_n\}$ be the set of given rectangles, where $r_i = (w_i, h_i)$. Let $W_\delta = \{r_i \mid w_i > 1 - \delta\}$ be the set of so-called δ-*wide* items and let $H_\delta = \{r_i \mid h_i > 1 - \delta\}$ be the set of δ-*high* items. To simplify the presentation, we denote the 1/2-wide items as *wide* items and the 1/2-high items as *high* items. Let W and H be the sets of wide and high items, respectively. The set of *small* items, i.e., items r_i with $w_i \leq 1/2$ and $h_i \leq 1/2$, is denoted by S. Finally, we call items that are wide and high at the same time *big*.

For a set T of items, let $\mathcal{A}(T) = \sum_{i \in T} w_i h_i$ be the total area and let $h(T) = \sum_{r_i \in T} h_i$ and $w(T) = \sum_{r_i \in T} w_i$ be the total height and total width, respectively. Finally, let $w_{\max}(T) = \max_{r_i \in T} w_i$ and $h_{\max}(T) = \max_{r_i \in T} h_i$.

Steinberg[13] proved the following theorem for his algorithm that we use as a subroutine.

Theorem 1 (Steinberg's algorithm). *If the following inequalities hold,*

$$w_{\max}(T) \leq a, \quad h_{\max}(T) \leq b, \text{ and } \quad 2\mathcal{A}(T) \leq ab - (2w_{\max}(T) - a)_+(2h_{\max}(T) - b)_+$$

where $x_+ = \max(x, 0)$, then it is possible to pack all items from T into $R = (a, b)$ in time $\mathcal{O}((n \log^2 n)/\log\log n)$.

Jansen & Zhang[10] showed the following corollary from this theorem.

Corollary 1. *If the total area of a set T of items is at most $1/2$ and there are no wide items (except a possible big item) then the items in T can be packed into a bin.*

Obviously, the corollary also holds for the case that there are no high items (except a possible big item). This corollary is an improvement upon Theorem 1 if there is a big item in T as in this case Theorem 1 would give a worse area bound.

Bansal, Caprara & Sviridenko[3] considered the two-dimensional knapsack problem in which each item $r_i \in I$ has an associated profit p_i and the goal is to maximize the total profit that is packed into a unit-sized bin. Using a very technical *Structural Lemma* they derived an algorithm that we call BCS algorithm in this paper. They showed the following theorem.

Theorem 2 (Bansal, Caprara & Sviridenko). *For any fixed $r \geq 1$ and $\delta > 0$, the BCS algorithm returns a packing of value at least $(1-\varepsilon)\mathrm{OPT}_{2KP}(I) - \varepsilon$ for instances I for which $p_i/\mathcal{A}(r_i) \in [1, r]$ for $r_i \in I$. The running time of the BCS algorithm is polynomial in the number of items.*

Here $\mathrm{OPT}_{2KP}(I)$ denotes the maximal profit that can be packed in a bin of unit size. In the case that $p_i = \mathcal{A}(r_i)$ we want to maximize the total packed area. Let $\mathrm{OPT}_{(a,b)}(T)$ denote the maximum area of items from T that can be packed into the rectangle (a, b), where individual items in T do not necessarily fit in (a, b). By appropriately scaling the bin, the items and the accuracy we get the following corollary.

Corollary 2 (Bansal, Caprara & Sviridenko). *For any fixed $\varepsilon > 0$, the BCS algorithm returns a packing of $I' \subseteq I$ in a bin of width $a \leq 1$ and height $b \leq 1$ such that $\mathcal{A}(I') \geq \mathrm{OPT}_{(a,b)}(I) - \varepsilon$.*

3 Our algorithm: Overview

As the asymptotic approximation ratio of the algorithm from Bansal, Caprara & Sviridenko[2] is arbitrarily close to $1.525\ldots$, there exists a constant k such that for any instance I with optimal value larger than k, their algorithm gives a solution of value at most $2\,\mathrm{OPT}(I)$. As we already mentioned in the introduction, this constant k is not explicitly known. We address the problem of approximating the problem within an absolute factor of 2, provided that the optimal value of the given instance is less than k. Combined with the algorithm from Bansal et al. we get an overall algorithm with an absolute approximation ratio of 2.

Our approach for packing instances I with $\mathrm{OPT}(I) \leq k$ consists of two parts. First, we give an algorithm that is able to pack instances I with $\mathrm{OPT}(I) = 1$ in two bins and second, we show how to approximate instances with $1 < \mathrm{OPT}(I) < k$ within a factor of 2. This at first glance surprising distinction is due to the inherent difficulty of packing wide and high items together into a single bin. In the case $\mathrm{OPT}(I) = 1$ we can not ensure a separation of the wide and high items into feasible sets whereas for $\mathrm{OPT}(I) > 1$ this is possible in most cases.

Organisation. In Section 4 we present the algorithm for packing instances that fit into one bin. In a first step we show how to pack these instances in two bins if the overall height of the δ-wide items is suitably bounded by a function in δ for some $\delta \in (\varepsilon, 1/2]$. If this is not the case we are able to derive substantial area guarantees for the sets of wide and high items. These area guarantees are utilized in the second step to pack all other instances I with $\mathrm{OPT}(I) = 1$.

The algorithm for instances I with optimal value within 2 and k is presented in Section 5. The basic algorithm consists of a large enumeration of items with area at least ε (for some constant $\varepsilon > 0$) and a separation of the wide and high items. On this basis we distinguish a total of four main cases that require different methods to be packed.

Finally, in Section 6 we put the different parts of the algorithm together.

4 Packing instances that fit into one bin

Throughout this section we assume that the given instance I can be packed into a single bin, i.e., $\mathrm{OPT}(I) = 1$. At first glance it seems surprising that packing such an instance into two bins is difficult. Still, we need to carefully analyse different cases to be able to give a polynomial-time algorithm that solves this problem.

Let $0 < \varepsilon < 1/200$ be a fixed constant. In a first step we consider instances that satisfy an upper bound for the total height of the δ-wide items for some $\delta \in (\varepsilon, 1/2]$. Using Steinberg's algorithm and the BCS algorithm we are able to pack these instances into two bins. In a second step we benefit from the upper bounds, as the instances that remain violate these bounds for *all* $\delta \in (\varepsilon, 1/2]$ and thus we have wide items of large total area. We use this area guarantee to give different methods to pack according to the total height of the wide items.

4.1 Small total height of the δ-wide items

We want to derive a packing of I into two bins in the case that the total height of the δ-wide items is small relative to δ, i.e., $h(W_\delta) \leq (\delta - \varepsilon)/(1 + 2\delta)$ for some $\delta \in (\varepsilon, 1/2]$.

For the ease of presentation let $\gamma = (\delta - \varepsilon)/(1 + 2\delta) \leq 1/4$ and assume that $h(W_\delta) \leq \gamma$. In a first step, we show that an almost optimal solution with a special structure exists. This special structure consists of a part of width $w(H_\gamma)$ for the γ-high items and a part of width $1 - w(H_\gamma)$ for the other items. The following lemma shows that almost all other items can be packed.

Lemma 1. *We have* $\mathrm{OPT}_{(1-w(H_\gamma),1)}(I \setminus H_\gamma) \geq \mathcal{A}(I \setminus H_\gamma) - 2\gamma$*.*

Proof. Consider an optimal packing of I. Remove all items that are completely contained in the top or bottom γ-margin. After this step there is no item directly above or below any item of $H_\gamma = \{r_i \mid h_i > 1-\gamma\}$. Thus we can cut the remaining packing at the left and right side of any items from H_γ. These cuts partition the packing into parts which can be swapped without losing any further item. Move all items of H_γ to the left of the bin and move all other parts of the packing to the right. The total area of the removed items is at most 2γ and thus a total area of at least $\mathcal{A}(I \setminus H_\gamma) - 2\gamma$ fits into a bin of size $(1 - w(H_\gamma), 1)$. □

In the second step, we actually derive a feasible packing that is based on the structure described above. First pack H_γ into a stack of width $w(H_\gamma)$ at the left side of the first bin. Note that $w(H_\gamma) \leq 1$. This leaves an empty space of width $1 - w(H_\gamma)$ and height 1 at the right. We therefore apply the BCS algorithm on $I \setminus H_\gamma$ and a bin of size $(1 - w(H_\gamma), 1)$ using an accuracy of ε. Lemma 1 and Corollary 2 yield that at least a total area of $\mathcal{A}(I \setminus H_\gamma) - 2\gamma - \varepsilon$ is packed by the algorithm.

Let T be the set of remaining items. We have $\mathcal{A}(T) \leq 2\gamma + \varepsilon$. Pack the remaining δ-wide items, i.e., the items of $T \cap W_\delta$, in a stack at the bottom of the second bin. Let $h' = h(T \cap W_\delta)$ be the total height of these items. We have $h' \leq h(W_\delta) \leq \gamma$. Let $T' = T \setminus W_\delta$ be the set of remaining items after this step. Then for $r_i \in T'$ with $r_i = (w_i, h_i)$ we have $w_i \leq 1 - \delta$ and $h_i \leq 1 - \gamma \leq 1 - h'$. The total area of the remaining items is

$$\mathcal{A}(T') \leq \mathcal{A}(T) - (1-\delta)h' \leq 2\gamma + \varepsilon - h' + \delta h'.$$

We pack these items with Steinberg's algorithm into the free rectangle of size (a, b) with $a = 1$ and $b = 1 - h'$ at the top of the second bin. In the following we show that this is possible by verifying the conditions of Theorem 1. We already stated that $w_{\max}(T') \leq 1 - \delta \leq 1$ and $h_{\max}(T') \leq 1 - \gamma \leq 1 - h'$. To show $2\mathcal{A}(T') \leq ab - (2w_{\max} - a)_+(2h_{\max} - b)_+$ we use

$$2\mathcal{A}(T') \leq 2(2\gamma + \varepsilon - h' + \delta h')$$

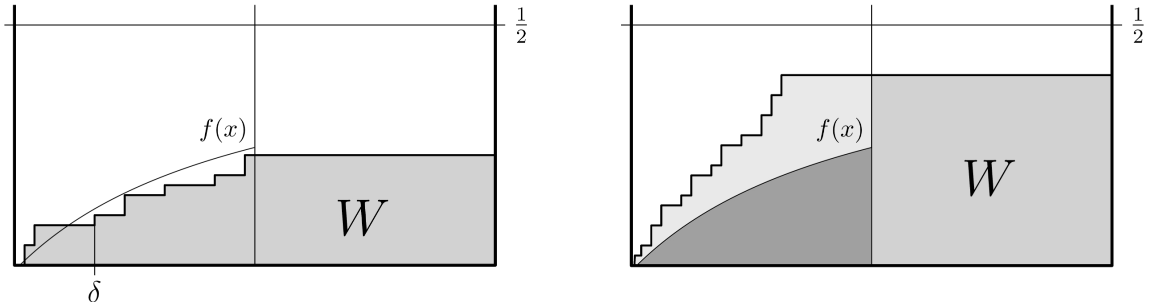
and (since $\delta \leq 1/2$ and therefore $\gamma \leq 1/4$)

$$\begin{aligned} ab - (2w_{\max} - a)_+(2h_{\max} - b)_+ &\geq 1 - h' - (2(1-\delta) - 1)_+(2(1-\gamma) - (1-h'))_+ \\ &= 1 - h' - (1 - 2\delta)_+(1 - 2\gamma + h')_+ \\ &= 1 - h' - (1 - 2\delta - 2\gamma + 4\delta\gamma + h' - 2\delta h') \\ &= 2(\delta + \gamma - 2\delta\gamma - h' + \delta h'). \end{aligned}$$

Starting with the definition of γ we get

$$\begin{aligned} && \gamma &= \frac{\delta - \varepsilon}{1 + 2\delta} \\ \Leftrightarrow && \gamma(1 + 2\delta) &= \delta - \varepsilon \\ \Leftrightarrow && \gamma + \varepsilon + 2\delta\gamma &= \delta \\ \Leftrightarrow && 2\gamma + \varepsilon - h' + \delta h' &= \delta + \gamma - 2\delta\gamma - h' + \delta h' \\ \Rightarrow && 2\mathcal{A}(T') &\leq ab - (2w_{\max} - a)_+(2h_{\max} - b)_+. \end{aligned}$$

So far we assumed the knowledge of $\delta \in (\varepsilon, 1/2]$ for which $h(W_\delta) \leq (\delta - \varepsilon)/(1 + 2\delta)$. It is easy to see that this value can be computed by calculating $h(W_\delta)$ for $\delta = 1 - w_i$ for all $r_i = (w_i, h_i)$ with $w_i > 1/2$. As $h(W_\delta)$ changes only for these values of δ, we will necessarily find a suitable δ if one exists. We therefore have the following lemma.



(a) If there exists a $\varepsilon < \delta \le 1/2$ such that $H(W_\delta) \le (\delta - \varepsilon)/(1 + 2\delta)$ then use Lemma 2 to pack the instance into two bins.

(b) Otherwise we get $\mathcal{A}(W) \ge \xi + h(W)/2$ for $\xi = 0.075$.

Fig. 1. Deriving the lower bound for the area of the wide items.

Lemma 2. *For any fixed* $\varepsilon > 0$*, there exists a polynomial-time algorithm that, given an instance* I *with* $\mathrm{OPT}(I) = 1$ *and* $h(W_\delta) \le (\delta - \varepsilon)/(1 + 2\delta)$ *for some* $\delta \in (\varepsilon, 1/2]$*, returns a packing of* I *into two bins.*

4.2 Using an area bound for the wide and high items

In this section we describe how to derive a guarantee on the total area of the wide and high items for the instances that cannot be solved by Lemma 2. Consider a unit-sized bin with the lower left corner at the origin of a cartesian coordinate system and consider the stack of wide items ordered by non-increasing width and aligned with the lower right corner of the bin. If there exists a $\delta \in (\varepsilon, 1/2]$ such that $h(W_\delta) \le (\delta - \varepsilon)/(1 + 2\delta)$ then we use the algorithm of Lemma 2 to pack the instance into two bins–see Figure 1(a). Otherwise the stack of wide items exceeds the function $f(x) = (x - \varepsilon)/(1 + 2x)$ for $x \in (\varepsilon, 1/2]$–see Figure 1(b). Then for $\varepsilon \le 1/200$ and $\xi = 0.075$ the total area of the wide items is

$$\begin{aligned}\mathcal{A}(W) &\ge \int_{\varepsilon}^{1/2} \frac{x - \varepsilon}{1 + 2x} dx + \frac{h(W)}{2} \\ &\ge \left[\frac{1}{4}\Big(2x - (2\varepsilon + 1)\log(2x + 1) + 1\Big)\right]_{\varepsilon}^{1/2} + \frac{h(W)}{2} \\ &\ge \xi + \frac{h(W)}{2}.\end{aligned}$$

Furthermore, for $\delta = 1/2$ we have

$$h(W) > \frac{1/2 - \varepsilon}{1 + 2 \cdot 1/2} = \frac{1}{4} - \frac{\varepsilon}{2}. \tag{1}$$

Obviously, we can apply Lemma 2 to the high items instead of the wide items as well. We get the following lemma.

Lemma 3. *For any input which cannot be packed by the methods described above, we have*

$$\mathcal{A}(W \cup H) \ge 2\xi + \frac{w(H) + h(W)}{2}.$$

It is crucial for our work that we get this additional area guarantee of $2\xi = 0.15$ on top of the trivial guarantee of $w(H)/2 + h(W)/2$ in the previous lemma.

Let us recapitulate the general idea of the algorithm. If $h(W_\delta) \leq (\delta - \varepsilon)/(1 + 2\delta)$ or $w(H_\delta) \leq (\delta - \varepsilon)/(1 + 2\delta)$ for some $\delta \in (\varepsilon, 1/2]$ we pack I in two bins using the methods from Lemma 2. Otherwise we rotate the instance such that we have $h(W) \geq w(H)$ and use the area guarantee of Lemma 3 to apply different methods for $w(H) > 1/2$ and $w(H) \leq 1/2$. In all cases we are able to pack into at most two bins. Before we can give the main lemmas to solve both cases above we need some preparations.

In the following we show the existence of a packing of the wide items and of high items with at least half of their total width of the high items with a nice structure, i.e., such that the wide and the high items are packed in stacks in different corners of the bin. We later use this observation to approximate such a packing.

Lemma 4. *For sets W and H of wide and high items with* $\mathrm{OPT}(W \cup H) = 1$ *there exists a packing of $W \cup H^*$ with $H^* \subseteq H$ and $w(H^*) \geq w(H)/2$ such that the wide items are stacked in the bottom right and the items from H^* are stacked in the top left corner of the bin.*

Proof. Consider a packing of wide items W and high items H into a bin. Associate each high item with the nearest side of the bin (an item that has the same distance to both sides of the bin can be associated with an arbitrary side). Assume w.l.o.g. that the total width of the items associated with the left is at least as large as the total width of the items associated with the right. Remove the items that are associated with the right and denote the other high items by H^*. Push the items of H^* together into a stack that is aligned with the left side of the bin by moving them purely horizontal and move the wide items such that they are aligned with the right side of the bin and form stacks at the bottom and top side of the bin. Order the stacks of the wide items by non-increasing order of width and the stack of the high items by non-increasing order of height. See left part of Figure 2.

Now apply the following process–see also Figure 2. Take the shortest item with respect to the width from the top stack of the wide items and insert it at the correct position into the bottom stack, i.e., such that the stack remains in the order of non-increasing width. Move the high items upwards if this insertion causes an overlap. Obviously this process moves all wide items to the bottom and retains a feasible packing. In the end, all wide items form a stack in the bottom right corner of the bin. Move the high items upwards such that they form a stack in the top left corner of the bin. □

We use this lemma to find a packing of the wide items together with high items of almost half their total width.

Lemma 5. *For any fixed $\varepsilon > 0$, there exists a polynomial-time algorithm that, given sets W and H of wide and high items with* $\mathrm{OPT}(W \cup H) = 1$*, returns a packing of $W \cup H'$ into a bin with $H' \subseteq H$ and $w(H') > w(H)/2 - \varepsilon$.*

Proof. By Lemma 4, a packing of $W \cup H^*$ exists. Let $H^*_{\geq\varepsilon} = \{r_i \in H^* \mid w_i \geq \varepsilon\}$ and $H^*_{<\varepsilon} = \{r_i \in H^* \mid w_i < \varepsilon\}$. We approximate $H^* = H^*_{\geq\varepsilon} \cup H^*_{<\varepsilon}$ as follows.

First, pack the items of W in a stack by non-increasing order of width and align this stack with the bottom right corner of the bin. Second, guess the set $H^*_{\geq\varepsilon}$. By guessing we mean that we enumerate all subsets of $\{r_i \in H \mid w_i \geq \varepsilon\}$ (which is possible as $|\{r_i \in H \mid w_i \geq \varepsilon\}| \leq 1/\varepsilon$)

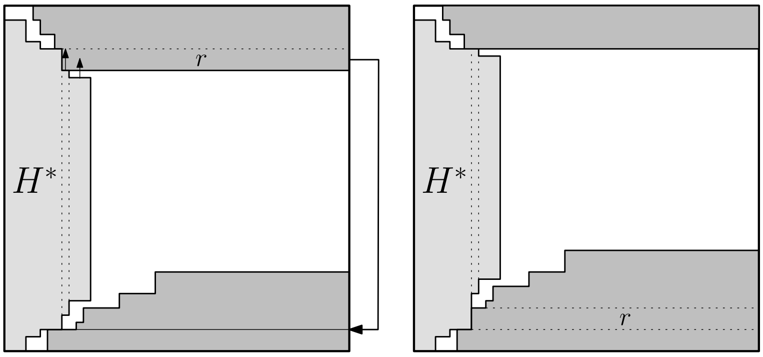


Fig. 2. Inserting item r into the bottom stack and moving the high items upwards

and apply the remainder of this algorithm on all these sets. As we eventually consider $H^*_{\geq\varepsilon}$, we assume that we can guess this set. Pack $H^*_{\geq\varepsilon}$ into a stack by non-increasing order of height and align this stack with the top left corner of the bin. Third, we approximate $H^*_{<\varepsilon}$ by greedily inserting items from $H_{<\varepsilon} = \{r_i \in H \mid w_i < \varepsilon\}$ into this stack. To do this, start with $H' = H^*_{\geq\varepsilon}$. Now sort the items of $H_{<\varepsilon}$ by non-increasing order of height and for each item try to insert it into the stack (at the correct position to preserve the order inside the stack). If this is possible, the item is added to H'.

Assume that $H_{<\varepsilon} = \{r_1, \ldots, r_m\}$ with $h_1 \leq \cdots \leq h_m$. Let $v_i = w(\{r_j \in H' \mid h_j \geq h_i\})$ and $v_i^* = w(\{r_j \in H^* \mid h_j \geq h_i\})$. Whenever an item r_i is not inserted in the stack we have $v_i > v_i^* - w_i$. To see this, assume that $v_i^* \geq v_i + w_i$. This means that the substack from H^* of items of height at least h_i has width larger than the substack of items of height at least h_i from the stack of H' *plus* the width of r_i. Thus r_i does not cause a conflict. Now it is easy to see by induction that $w(H') > w(H^*) - \varepsilon$ at the end. □

With these preparations, the following lemma is easy to show.

Lemma 6. *Let $\varepsilon > 0$ and let I be an instance with* $\mathrm{OPT}(I) = 1$*, $h(W) \geq w(H) > 1/2$, and $h(W_\delta) > (\delta - \varepsilon)/(1 + 2\delta)$ and $w(H_\delta) > (\delta - \varepsilon)/(1 + 2\delta)$ for all $\delta \in (\varepsilon, 1/2]$. There exists a polynomial-time algorithm that returns a packing of I into two bins.*

Proof. Use Lemma 5 to pack $W \cup H'$ with $H' \subseteq H$ and $w(H') > w(H)/2 - \varepsilon$ in the first bin. Build a stack of the remaining high items $H \setminus H'$ and align it with the left side of the second bin. The width of this stack is $w(H \setminus H') < w(H)/2 + \varepsilon$. Note that $w(H \setminus H') \leq 1/2$, as otherwise $h(W) \geq w(H) \geq 1 - 2\varepsilon$, which gives a contradiction since by Lemma 3 we would then have $\mathcal{A}(W \cup H) \geq 2\xi + (w(H) + h(W))/2 \geq 2\xi + 1 - 2\varepsilon > 1$. Pack the remaining items T with Steinberg's algorithm in the free rectangle of size (a, b) with $a = 1 - w(H \setminus H')$ and $b = 1$ to the right of the second bin. This is possible since $w_{\max}(T) \leq 1/2 \leq 1 - w(H \setminus H')$,

$h_{\max}(T) \leq 1/2$ and with Lemma 3 we have

$$\begin{aligned}
2\mathcal{A}(T) &\leq 2\Big(1 - 2\xi - \frac{w(H)+h(W)}{2}\Big) \\
&\leq 2 - 4\xi - \frac{w(H)}{2} - \frac{w(H)}{2} - \frac{1}{2} && \text{as } h(W) \geq 1/2 \\
&< \frac{3}{2} - 4\xi - w(H \setminus H') + \varepsilon - \frac{1}{4} && \text{as } \frac{w(H)}{2} \geq w(H \setminus H') - \varepsilon \text{ and } \frac{w(H)}{2} > \frac{1}{4} \\
&< 1 - w(H \setminus H') && \text{as } 4\xi > \frac{1}{4} + \varepsilon \\
&= ab - (2w_{\max} - a)_+(2h_{\max} - b)_+ && \text{as } 2h_{\max} - b \leq 0.
\end{aligned}$$

□

In the following we assume that $w(H) \leq 1/2$ as otherwise we could pack the instance into two bins with the algorithms of Lemma 2 or Lemma 6. Furthermore, we still have our initial assumption $h(W) \geq w(H)$. The following lemma shows that certain sets of small items can be packed together with the set of wide items.

Lemma 7. *Any set $T = \{r_1, \ldots, r_m\}$ where $r_i = (w_i, h_i)$ with $w_i \leq 1/2$, $h_i \leq 1 - h(W)$ for $i = 1, \ldots, m$ and total area $\mathcal{A}(T) \leq 1/2 - h(W)/2$ can be packed together with W.*

Proof. Pack W into a stack of height $h(W)$ and align this stack with the bottom of the bin. Use Steinberg's algorithm to pack T into the free rectangle of size (a, b) with $a = 1$ and $b = 1 - h(W)$ above W. This is possible since $w_{\max}(T) \leq 1/2$, $h_{\max}(T) \leq 1 - h(W)$ and $2\mathcal{A}(T) \leq 1 - h(W) = ab = ab - (2w_{\max} - a)_+(2h_{\max} - b)_+$ as $2w_{\max} - a \leq 0$. □

Obviously, Lemma 7 can also be formulated such that we pack the high items together with a set of small items of total area at most $1/2 - w(H)/2$ (in this case we do not need a condition like $h_i \leq 1 - h(W)$, as $w(H) \leq 1/2$ and thus all remaining items fit into the free rectangle next to the stack of H). This suggests partitioning the small items into sets with these area bounds in order to pack them with the wide and high items. Before we can do this, we consider two cases where we have to apply a different packing. In all cases, we pack W in a stack of height $h(W)$ in the bottom right corner of the first bin and pack H in a stack of width $w(H)$ in the top left corner of the second bin.

Let ω be the greatest width in the stack of W that is packed above height $1/2$ in this packing (let $\omega = 1/2$ if $h(W) \leq 1/2$)–see Figure 3. We consider the set $\widetilde{H} = \{r_i \mid h_i \in (1 - h(W), 1/2]\}$, i.e., the set of remaining items that do not fit above the stack of the wide items (and thus violate the condition of Lemma 7). Since $\widetilde{H} = \emptyset$ for $h(W) \leq 1/2$, the following case can only occur if $h(W) > 1/2$.

Case 1. $w(\widetilde{H}) \geq (1 - \omega)/2$.
If there is an item $r_i = (w_i, h_i) \in \widetilde{H}$ with $w_i > (1 - \omega)/2$ then we pack this item in the top left corner of the first bin. It is easy to see that this is possible since $\mathrm{OPT}(I) = 1$. Otherwise greedily pack items from $\widetilde{H}$ into a horizontal stack in the top left corner of the first bin as long as they fit. Since all items in $\widetilde{H}$ have width at most $(1 - \omega)/2$ we can pack a total width of at least $(1 - \omega)/2$ before a conflict occurs.

In both cases we packed a total area of at least $(1 - h(W))(1 - \omega)/2$ from $\widetilde{H}$ into the first bin–see rectangle R_1 in Figure 3. Furthermore, we can use the definition of ω to improve

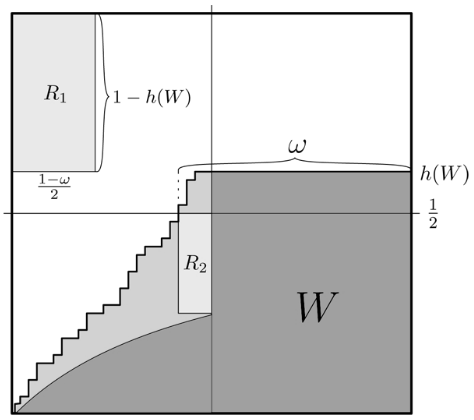


Fig. 3. The greatest width in the stack of W that is packed above height $1/2$ is denoted by ω. The additional area guarantees for Case 1 of $\mathcal{A}(R_1) = (1-h(W))(1-\omega)/2$ and $\mathcal{A}(R_2) = (\omega - 1/2)/4$ are drawn with lighter shading. The original area guarantee from Lemma 3 is drawn with darker shading.

the estimate from Lemma 3. In Lemma 3 all wide items that are packed above height $1/4$ only contribute with their trivial area guarantee of half their height. Now we know that the items that are packed between height $1/4$ and $1/2$ have width at least ω and this gives an additional area of at least $(\omega - 1/2)\cdot 1/4$–see rectangle R_2 in Figure 3. Thus we packed an overall area of

$$\begin{aligned} A &\geq 2\xi + \frac{w(H)+h(W)}{2} + \frac{\omega - 1/2}{4} + \frac{(1-h(W))(1-\omega)}{2} \\ &= \frac{3}{8} + 2\xi + \frac{w(H)}{2} + \omega\Big(\frac{h(W)}{2} - \frac{1}{4}\Big) \\ &\geq \frac{3}{8} + 2\xi + \frac{w(H)}{2} && \text{as } h(W) \geq 1/2 \text{ and } \omega > 0. \end{aligned}$$

Therefore, the remaining items have total area at most $5/8 - 2\xi - w(H)/2 < 1/2 - w(H)/2$. Thus Lemma 7 allows us to pack these items together with the high items in the second bin.

Let r', r'' be the two largest items in $S \setminus \widetilde{H}$, i.e., among the remaining small items S with $h_i \leq 1 - h(W)$.

Case 2. $\mathcal{A}(\{r', r''\}) \geq 1/2 - 2\xi - h(W)/2$.
Pack r' in the top left and r'' in the top right corner of the first bin. This is possible as the width of both items is at most $1/2$ and the height is at most $1-h(W)$ for r', r''. By Lemma 3 and the condition for this case the total area of the packed items is

$$A \geq 2\xi + \frac{w(H)+h(W)}{2} + \frac{1}{2} - 2\xi - \frac{h(W)}{2} \geq \frac{1}{2} + \frac{w(H)}{2}.$$

Again we can use the method of Lemma 7 to pack the remaining items together with H in the second bin.

Case 3. Otherwise we have $w(\widetilde{H}) < (1-\omega)/2$ and $\mathcal{A}(\{r', r''\}) < 1/2 - 2\xi - h(W)/2$. This yields that $\mathcal{A}(\widetilde{H}) < (1-\omega)/4$ and $\mathcal{A}(\{r''\}) < 1/4 - \xi - h(W)/4$ (where we assume that $\mathcal{A}(\{r'\}) \geq \mathcal{A}(\{r''\})$. Use the following greedy algorithm to partition the remaining items into two sets that will be packed together with W and H using Lemma 7.

1. Create sets S_1 and S_2 with capacities $c_1 = 1/2 - h(W)/2$ and $c_2 = 1/2 - w(H)/2$, respectively,
2. add r' to S_1 and add all items of $\widetilde{H}$ to S_2,
3. take the remaining items by non-increasing order of size and greedily add them to the set of greater remaining free capacity, i.e., to a set with maximal $c_i - \mathcal{A}(S_i)$.

In the following we show that $\mathcal{A}(S_1) \leq c_1$ and $\mathcal{A}(S_2) \leq c_2$. First note that this holds after Step 2 since

$$\begin{aligned} \mathcal{A}(\{r'\}) &< \frac{1}{2} - 2\xi - \frac{h(W)}{2} < \frac{1}{2} - \frac{h(W)}{2} = c_1 && \text{and} \\ \mathcal{A}(\widetilde{H}) &< \frac{1-\omega}{4} < \frac{1-\omega}{2} \leq \frac{1}{2} - \frac{w(H)}{2} = c_2 && \text{as } \omega \geq 1/2 \geq w(H). \end{aligned}$$

Assume that after Step 3 one of the sets has total area greater than its corresponding capacity. Then there is an item r^* that has been added in Step 3 and that violates the capacity for the first time. Since $h(W) \geq 1/4 - \varepsilon/2$ by Inequality (1) and $\varepsilon \leq 1/4$ we have $\mathcal{A}(\{r^*\}) \leq \mathcal{A}(\{r''\}) \leq 1/4 - \xi - h(W)/4 \leq 3/16 - \xi + \varepsilon/8 < 0.15$. Assume w.l.o.g. that r^* was added to S_1. Then we have $\mathcal{A}(S_1) > c_1$ and $\mathcal{A}(S_2) \geq c_2 - \mathcal{A}(r^*)$ as otherwise r^* would have been added to S_2. Thus we have

$$\mathcal{A}(S_1) + \mathcal{A}(S_2) > c_1 + c_2 - \mathcal{A}(\{r^*\}) \geq c_1 + c_2 - 0.15.$$

With Lemma 3 we get the contradiction

$$\begin{aligned} \mathcal{A}(S_1) + \mathcal{A}(S_2) &\leq 1 - \mathcal{A}(W \cup H) \\ &\leq 1 - 2\xi - \frac{w(H) + h(W)}{2} \\ &= c_1 + c_2 - 2\xi \\ &= c_1 + c_2 - 0.15. \end{aligned}$$

Thus both sets do not violate their capacities and we can use the methods of Lemma 7 to pack S_1 together with W in the first bin and S_2 together with H in the second bin. We showed the following lemma.

Lemma 8. *Let $\varepsilon > 0$ and let I be an instance with* $\mathrm{OPT}(I) = 1$, $w(H) \leq 1/2$, *and* $h(W_\delta) > (\delta - \varepsilon)/(1+2\delta)$ *and* $w(H_\delta) \leq (\delta-\varepsilon)/(1+2\delta)$ *for all* $\delta \in (\varepsilon, 1/2]$. *There exists a polynomial-time algorithm that returns a packing of I into two bins.*

This concludes our algorithm for instances I with $\mathrm{OPT}(I) = 1$ as one of the Lemmas 2, 6 and 8 can pack I into two bins. In the next section we show how to handle instances with $\mathrm{OPT}(I) > 1$.

5 Packing instances that fit into a constant number of bins

In the following we describe our algorithm that packs the instances I with $2 \leq \mathrm{OPT}(I) < k$ into $2\mathrm{OPT}(I)$ bins. Let $\varepsilon = 1/(40k^3 + 2)$.

Let $L = \{r_i \mid w_i h_i > \varepsilon\}$ be the set of *large* items and let $T = \{r_i \mid w_i h_i \leq \varepsilon\}$ be the set of *tiny* items. As defined in Section 2 we refer to items as wide (W), high (H), small (S) and big, according to their side lengths. Note that the terms *large* and *tiny* refer to the area of the items whereas the predicates *big*, *wide*, *high* and *small* are given depending on the width and height of the items. Also note that, e.g., an item can be tiny and high, or wide and big at the same time.

We guess $\ell = \mathrm{OPT}(I)$ and open 2ℓ bins that we denote by $B_1, \ldots, B_\ell$ and $C_1, \ldots, C_\ell$. By *guessing* we mean that we iterate over all possible values for ℓ and apply the remainder of this algorithm on every value. As there are only a constant number of values, this is possible in polynomial time. We assume that we know the correct value of ℓ as we eventually consider this value in an iteration. For the ease of presentation, we also denote the sets of items that are associated with the bins by $B_1, \ldots, B_\ell$ and $C_1, \ldots, C_\ell$. We will ensure that the set of items that is associated with a bin is feasible and a packing is known or can be computed in polynomial time.

Let I_i^* be the set of items in the i-th bin in an optimal solution. We assume w.l.o.g. that $\mathcal{A}(I_i^*) \geq \mathcal{A}(I_j^*)$ for $i < j$. Then we have

$$\mathcal{A}(I) = \mathcal{A}(I_1^*) + \cdots + \mathcal{A}(I_\ell^*) \leq \ell \cdot \mathcal{A}(I_1^*) \tag{2}$$

In a first step, we guess the assignment of the large items to bins. Using this assignment and the BCS algorithm we pack a total area of at least $\mathcal{A}(I_1^*) - \varepsilon$ into B_1 and keep C_1 empty. This step has the purpose of providing a good area bound for the first bin (especially since it is set into relation to the fullest bin of an optimal solution in Inequality (2)) and leaving a free bin for later use. We use a slightly technical definition (which is given later in the details) for the profits of the items to ensure that the large items that are assigned to B_1 are actually packed. For all other bins we reserve B_i for the wide and small items (except the big items) and C_i for the high and big items for $i = 2, \ldots, \ell$. This separation enables us to use Steinberg's algorithm to pack up to half of the bins' area. In detail, the first part of the algorithm works as follows.

1. Guess $L_i = I_i^* \cap L$ for $i = 1, \ldots, \ell$.
2. Apply the BCS algorithm on $L_1 \cup T$ with $p_i = \mathcal{A}(r_i)(1/\varepsilon + 1)$ for $r_i \in L_1$ and $p_i = \mathcal{A}(r_i)$ for $r_i \in T$ using an accuracy of $\varepsilon^2/(1 + 2\varepsilon)$. Assign the output to bin B_1 and keep an empty bin C_1.
3. For $i = 2, \ldots, \ell$, assign the wide and small items of L_i to B_i (omitting big items) and assign the high and big items of L_i to C_i. That is $B_i = L_i \setminus H$ and $C_i = L_i \cap H$.
4. For $i = 2, \ldots, \ell$, greedily add tiny wide items from $T \cap W$ by non-increasing order of width to B_i as long as $\mathcal{A}(B_i) \leq 1/2$ and greedily add tiny high items from $T \cap H$ by non-increasing order of height to C_i as long as $w(C_i) \leq 1$.

Corollary 1 shows that using Steinberg's algorithm the bins $B_2, \ldots, B_\ell$ can be packed as there are no wide items and the total area is at most $1/2$. The bins $C_2, \ldots, C_\ell$ can be packed with a simple stack as they contain only high items of total width at most 1. Observe that in Step 4 we only add to a new bin B_i if the previous bins contain items of total area at least $1/2 - \varepsilon$ and we only add to a new bin C_i if the previous bins contain items of total width at least $1 - 2\varepsilon$ (as the width of the tiny high items is at most 2ε) and thus of total area at least $1/2(1 - 2\varepsilon) = 1/2 - \varepsilon$. After the application of this first part of the algorithm, some tiny items $T' \subseteq T$ might remain unpacked. Note that if $\mathcal{A}(B_\ell) < 1/2 - \varepsilon$, then there are no wide items in T' and if $\mathcal{A}(C_\ell) < 1/2 - \varepsilon$ then there are no high items in T' (as these items would have been packed in Step 4). We distinguish different cases to continue the packing according to the filling of the last bins B_ℓ and C_ℓ. Prior to this we prove the area bound for B_1 that we already mentioned above.

First note that Theorem 2 can be applied as $p_i/\mathcal{A}(r_i) \in \{1, 1/\varepsilon + 1\}$ for all items in $L_1 \cup T$. Recall that we used the technical definition of the profits to ensure that L_1 is completely packed, i.e., $B_1 \cap L = I_1^* \cap L$. This is easy to see since $p_i > 1 + \varepsilon$ for $r_i \in L_1$, whereas $p(\widetilde{T}) = \mathcal{A}(\widetilde{T}) \leq 1$ for any feasible set $\widetilde{T} \subseteq T$. Furthermore, for the set of packed tiny items $B_1 \cap T$ we have

$$\mathcal{A}(B_1 \cap T) \geq \mathcal{A}(I_1^* \cap T) - \varepsilon$$

since $(1/\varepsilon + 1)\mathcal{A}(B_1 \cap L) + \mathcal{A}(B_1 \cap T) = p(B_1)$ and

$$\begin{aligned}
p(B_1) &\geq \Big(1 - \frac{\varepsilon^2}{1+2\varepsilon}\Big)\mathrm{OPT}(L_1 \cup T) - \frac{\varepsilon^2}{1+2\varepsilon} && \text{by Theorem 2}\\
&\geq \Big(1 - \frac{\varepsilon^2}{1+2\varepsilon}\Big)\Big[\Big(\frac{1}{\varepsilon}+1\Big)\mathcal{A}(I_1^* \cap L) + \mathcal{A}(I_1^* \cap T)\Big] - \frac{\varepsilon^2}{1+2\varepsilon}\\
&= \Big(\frac{1}{\varepsilon}+1\Big)\mathcal{A}(I_1^* \cap L) + \mathcal{A}(I_1^* \cap T) - \Big(\frac{1}{\varepsilon}+2\Big)\frac{\varepsilon^2}{1+2\varepsilon}\\
&= \Big(\frac{1}{\varepsilon}+1\Big)\mathcal{A}(B_1 \cap L) + \mathcal{A}(I_1^* \cap T) - \varepsilon.
\end{aligned}$$

Thus we have

$$\mathcal{A}(B_1) \geq \mathcal{A}(I_1^*) - \varepsilon. \tag{3}$$

Now we are ready to start with the case analysis.

Case 1. $\mathcal{A}(B_\ell) < 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) < 1/2 - \varepsilon$.
In this case T' does not contain any wide or high items as these items would have been packed to B_ℓ or C_ℓ. Greedily add items from T' into all bins except B_1 as long as the bins contain items of total area at most 1/2. After adding the items from T', either all items are assigned to a bin (and can thus be packed) or each bin contains items of total area at least $1/2 - \varepsilon$ and we packed a total area of at least

$$\begin{aligned}
A &\geq \mathcal{A}(B_1) + (2\ell - 1)\Big(\frac{1}{2} - \varepsilon\Big)\\
&\geq \mathcal{A}(I_1^*) + \ell - \frac{1}{2} - 2\ell\varepsilon && \text{from Inequality (3)}\\
&\geq \mathcal{A}(I_1^*) + \ell\mathcal{A}(I_1^*) + \ell(1 - \mathcal{A}(I_1^*)) - \frac{1}{2} - 2\ell\varepsilon\\
&\geq \ell\mathcal{A}(I_1^*) + \frac{1}{2} - 2\ell\varepsilon && \text{as } \ell \geq 1 \text{ and } 1 - \mathcal{A}(I_1^*) \geq 0\\
&> \ell\mathcal{A}(I_1^*) && \text{as } \varepsilon < \frac{1}{4\ell}.
\end{aligned}$$

Since this contradicts $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$ from Inequality (2), all items are packed.

Case 2. $\mathcal{A}(B_\ell) \geq 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) \geq 1/2 - \varepsilon$.
In this case T' might contain wide and high items. On the other hand the bin C_1 is still available for packing. We use the area of the items in bin C_ℓ to bound the total area of the

packed items. With a calculation as in Case 1 we get

$$\begin{aligned} A &\geq \mathcal{A}(B_1) + \mathcal{A}(C_\ell) + \overbrace{(2\ell-3)\Big(\frac{1}{2}-\varepsilon\Big)}^{\text{all bins except } B_1, C_1, C_\ell} \\ &\geq \ell\mathcal{A}(I_1^*) + \mathcal{A}(C_\ell) - \frac{1}{2} - (2\ell-2)\varepsilon. \end{aligned}$$

Again from Inequality (2) we have $\mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$. Thus we get

$$\mathcal{A}(T') \leq \mathcal{A}(I) - A \leq \frac{1}{2} + (2\ell-2)\varepsilon - \mathcal{A}(C_\ell) \quad \text{and hence} \tag{4}$$

$$\mathcal{A}(T') \leq (2\ell-1)\varepsilon \qquad\qquad \text{as } \mathcal{A}(C_\ell) \geq 1/2-\varepsilon \tag{5}$$

By Inequality (4) we know that if $\mathcal{A}(C_\ell) > 1/2 + (2\ell-2)\varepsilon$ then $\mathcal{A}(T') < 0$ and thus all items are packed. Therefore we assume that $\mathcal{A}(C_\ell) \leq 1/2 + (2\ell-2)\varepsilon$.

We consider the set $\widehat{H} = \{r_i \in C_\ell \mid h_i \leq 3/4\}$. If $w(\widehat{H}) \geq (4\ell-3)\varepsilon$ then remove $\widehat{H}$ from C_ℓ and pack it in a stack in C_1 instead. As we now have $\mathcal{A}(C_\ell) \leq 1/2 - (2\ell-1)\varepsilon$ and $\mathcal{A}(T' \setminus W) \leq \mathcal{A}(T') \leq (2\ell-1)\varepsilon$ by Inequality (5), we can pack $T' \setminus W$ together with C_ℓ. The remaining items $T' \cap W$ have total height at most $2(2\ell-1)\varepsilon$ and thus fit above $\widehat{H}$ into C_1.

Otherwise, there is no item $r' = (w', h')$ in C_ℓ with $h' \leq 3/4$ and $w' \geq (4\ell-3)\varepsilon$. Let $\widetilde{H} = \{r_i \in C_\ell \cup T' \mid h_i > 3/4\}$. Observe that we have

$$w(\widetilde{H}) \leq \frac{\mathcal{A}(C_\ell \cup T')}{3/4} \leq \frac{4}{3}\Big(\frac{1}{2} + (2\ell-2)\varepsilon + (2\ell-1)\varepsilon\Big) = \frac{2}{3} + \Big(\frac{16}{3}\ell - 4\Big)\varepsilon < 1.$$

We take *all* high items from $C_\ell \cup T'$ and order them by non-increasing height. Now pack the items greedily into a stack of width up to 1 and pack this stack into C_ℓ. We have $w(C_\ell) \geq 1 - (4\ell-3)\varepsilon$ as the total width of the items from $\widetilde{H}$ is bounded and thus all further items have width at most $(4\ell-3)\varepsilon$ (as otherwise $\widehat{H} \geq (4\ell-3)\varepsilon$ and we had solve the problem in the previous step). For the remaining items T' we have $h_{\max}(T') \leq 3/4$ and $\mathcal{A}(T') \leq 1/2 - (2\ell-2)\varepsilon - \mathcal{A}(C_\ell) \leq (4\ell - 7/2)\varepsilon \leq 1/4$ (by Inequality 4 and as $\mathcal{A}(C_\ell) \geq w(C_\ell)/2 \geq 1/2 - (2\ell-3/2)\varepsilon$ and $\varepsilon \leq 1/(16\ell)$. Thus T' can be packed into bin C_1 using Steinberg's algorithm.

Case 3. $\mathcal{A}(B_\ell) < 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) \geq 1/2 - \varepsilon$.
If $w(T' \cap H) \leq 1$ then pack $T' \cap H$ in C_1 and proceed as in Case 1.

The subcase where $w(T' \cap H) > 1$ is the most difficult of all four cases. The challenge that we face is that $w(H)$ can be close to ℓ (which is a natural upper bound) but we can only ensure a packed total width of at least $\ell(1-2\varepsilon)$ in the bins $C_1, \ldots, C_\ell$. So we have to pack high items into the bins $B_2, \ldots, B_\ell$. We distinguish two further subcases.

1. Assume that there exists $j \in \{2, \ldots, \ell\}$ with $w(L_j \cap H) > 10\ell\varepsilon$, i.e., the total width of the items that are large and high and associated with the i-th bin in an optimal packing is large enough such that moving these items away gives sufficient space for the still unpacked high items.
   Go back to Step 3 in the first part of the algorithm and omit separating the items from L_j. Instead we assign the items from L_j to bin B_j and keep C_j free at the moment. Note that L_j admits a packing into a bin as L_j corresponds to the large items in a bin of an optimal solution. Since $|L_j| \leq 1/\varepsilon$ we can find such a packing in constant time.

While greedily adding tiny items in Step 4, we skip B_j for the wide items and we continue packing high items in C_1 after we have filled $C_2, \ldots, C_\ell$. As we moved high items of total width at least $10\ell\varepsilon$ to B_j and we can pack high items of total width at least $1 - 2\varepsilon$ into each bin, no high items remains after this step. Finally, greedily add remaining tiny items to bins $B_2, \ldots, B_\ell$ except B_j, using the area bound $1/2$.

Now consider the bins C_1 and C_j. Both contain only tiny items, as we moved the large items from C_j to B_j. We packed the tiny items greedily by height and thus all items in C_j have height greater or equal to any item in C_1. Let h' be greatest height in C_1. Then we have $\mathcal{A}(C_j) \geq h'(1-2\varepsilon)$. Furthermore, we know that $w(H) > \ell(1-2\varepsilon)$. Thus we have

$$\mathcal{A}(H) > (\ell - 1)(1/2 - \varepsilon) + h'(1 - 2\varepsilon).$$

If after the modified Step 4 tiny items remain unpacked, then all bins B_i for $i \in \{2, \ldots, \ell\} \setminus \{j\}$ have area $\mathcal{A}(B_i) \geq 1/2 - \varepsilon$. By summing up the area of the high items separately we get a total packed area of at least

$$\begin{aligned}
A &\geq \mathcal{A}(B_1) + \overbrace{(\ell-2)\left(\frac{1}{2} - \varepsilon\right)}^{B_i \text{ for } i \in \{2,\ldots,\ell\} \setminus \{j\}} + \mathcal{A}(H) \\
&> \mathcal{A}(B_1) + (\ell-2)\left(\frac{1}{2} - \varepsilon\right) + (\ell-1)\left(\frac{1}{2} - \varepsilon\right) + h'(1-2\varepsilon) \\
&\geq \mathcal{A}(I_1^*) + \ell - \frac{3}{2} + h' - (2\ell - 2 + 2h')\varepsilon && \text{by Inequality (3)} \\
&\geq \mathcal{A}(I_1^*) + \ell\mathcal{A}(I_1^*) + \ell(1 - \mathcal{A}(I_1^*)) - \frac{3}{2} + h' - 2\ell\varepsilon && \text{as } h' \leq 1 \\
&\geq \ell\mathcal{A}(I_1^*) - \frac{1}{2} + h' - 2\ell\varepsilon && \text{as } \ell \geq 1 \text{ and } 1 - \mathcal{A}(I_1^*) \geq 0.
\end{aligned}$$

On the other hand we have $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$ by Inequality (2). Thus the total area of the remaining items T' is at most $\mathcal{A}(T') \leq 1/2 + 2\ell\varepsilon - h'$. If $h' \geq 1/2 + 2\ell\varepsilon$ we packed all items.

Otherwise we have $1/2 < h' < 1/2 + 2\ell\varepsilon$ and

$$\mathcal{A}(T') \leq 2\ell\varepsilon. \tag{6}$$

We will pack T' in C_1 together with the already packed high items. Observe that

$$w(C_1 \cap H) \leq 1 - 8\ell\varepsilon \tag{7}$$

as we move high items of total area of at least $10\ell\varepsilon$ to B_j and all bins $C_2, \ldots, C_\ell$ are filled up to a width of at least $1 - 2\varepsilon$.

We pack the remaining items T' into three rectangles $R_1 = (1, 4\ell\varepsilon)$, $R_2 = (8\ell\varepsilon, 1 - 4\ell\varepsilon)$ and $R_3 = (1 - 8\ell\varepsilon, 1/2 - 6\ell\varepsilon)$ which can be packed in C_1 together with $C_1 \cap H$ as follows– see Figure 4. Pack the stack of $C_1 \cap H$ in the lower left corner and pack R_3 above this stack. As $h' + h(R_3) \leq 1 - h(R_1)$, R_1 fits in the top of C_1. Finally, pack R_2 in the bottom right corner. This is possible as $h(R_2) \leq 1 - h(R_1)$ and $w(C_1 \cap H) \leq 1 - 8\ell\varepsilon = 1 - w(R_3)$ by Inequality (7).

Now pack $T' \cap W$ in a stack in R_1 (which is possible since $h(T' \cap W) \leq 2\mathcal{A}(T') \leq 4\ell\varepsilon$ by Inequality (6)) and pack all $r_i = (w_i, h_i) \in T'$ with $h_i > 1/2 - 6\ell\varepsilon$ in a vertical stack in R_2 (this fits as the total width of items with $h_i > 1/2 - 6\ell\varepsilon$ is at most $\mathcal{A}(T')/(1/2 - 6\ell\varepsilon) \leq 8\ell\varepsilon$

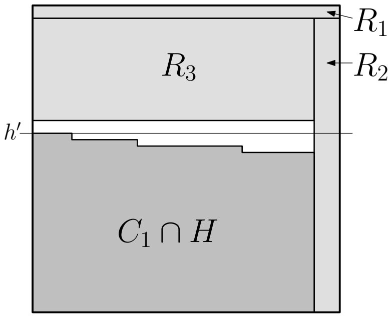


Fig. 4. The three rectangles R_1, R_2 and R_3 for packing T' together with $C_1 \cap H$ in C_1.

by Inequality (6) and as $6\ell\varepsilon \leq 1/4$). Finally, use Steinberg's algorithm to pack the remaining items in R_3. This is possible since $w_{\max} \leq 1/2$, $h_{\max} \leq 1/2 - 6\ell\varepsilon$ and

$$\begin{aligned}2\mathcal{A}(T') \leq 4\ell\varepsilon &\leq \frac{1}{2} - 14\ell\varepsilon + 96\ell^2\varepsilon^2\\ &= (1-8\ell\varepsilon)\Big(\frac{1}{2} - 6\ell\varepsilon\Big) - (1-1+8\ell\varepsilon)_+\Big(1 - 12\ell\varepsilon - \frac{1}{2} + 6\ell\varepsilon\Big)_+.\end{aligned}$$

This finishes the first case where we assumed that there exists $j \in \{2, \dots, \ell\}$ with $w(L_j \cap H) > 10\ell\varepsilon$.

2. Now assume that we have $w(L_j \cap H) \leq 10\ell\varepsilon$ for all $j \in \{2, \dots, \ell\}$ and in particular $w_i \leq 10\ell\varepsilon$ for all items $r_i = (w_i, h_i) \in (L_2 \cup \dots \cup L_\ell)$. Thus all high items that are not packed in B_1 are thin, i.e., have width at most $10\ell\varepsilon$. We use this fact by repacking the high items greedily by non-increasing height in the bins $C_1, \dots, C_\ell$. Each bin contains high items of total width at least $1 - 10\ell\varepsilon$ afterwards. Thus high items of total width at most $10\ell^2\varepsilon$ remain unpacked. This is worse than in the previous case but since we repacked all items we can get a nice bound on the height of the unpacked items. Let h' be the smallest height in C_ℓ. Then all items in $C_1, \dots, C_\ell$ have height at least h' and the remaining items T' have height at most h'. If there is an $i \in \{2, \dots, \ell\}$ with $\mathcal{A}(B_i) \leq 1 - h' - 10\ell^2\varepsilon$ then we can add the remaining items T' to B_i using Steinberg's algorithm. To see this note that $h_{\max}(T' \cup B_i) \leq h'$ and $2\mathcal{A}(T' \cup B_i) \leq 2\mathcal{A}(B_i) + 2(10\ell^2\varepsilon)h' \leq 2 - 2h'$ which corresponds to the bound from Theorem 1.
Otherwise for all $i \in \{2, \dots, \ell\}$ we have $\mathcal{A}(B_i) \geq 1 - h' - 0\ell^2\varepsilon$. Then we packed a total area of at least

$$\begin{aligned}A &\geq \mathcal{A}(B_1) + \mathcal{A}(C_1 \cup \dots \cup C_\ell) + \mathcal{A}(B_2 \cup \dots \cup B_\ell)\\ &\geq \mathcal{A}(B_1) + h'\ell(1 - 10\ell\varepsilon) + (\ell - 1)(1 - h' - 10\ell^2\varepsilon)\\ &\geq \mathcal{A}(I_1^*) + \ell - 1 + h' - (10\ell^2 + 10\ell^2 + 10\ell^3 + 1)\varepsilon\\ &> \mathcal{A}(I_1^*) + \ell\mathcal{A}(I_1^*) + \ell(1 - \mathcal{A}(I_1^*)) - 1 + \frac{1}{2} - (20\ell^3 + 1)\varepsilon \quad \text{as } \ell \geq 2 \text{ and } h' > \frac{1}{2}\\ &\geq \ell\mathcal{A}(I_1^*) + \frac{1}{2} - (20\ell^3 + 1)\varepsilon \quad \text{as } \ell \geq 1 \text{ and } 1 - \mathcal{A}(I_1^*) \geq 0.\end{aligned}$$

And since $1/2 - (20\ell^3 + 1)\varepsilon \geq 0$ and $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$, no item remains unpacked.

Thus in both subcases we were able to derive a feasible packing.

Case 4. $\mathcal{A}(B_\ell) \geq 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) < 1/2 - \varepsilon$.
In this case T' contains no high items. If there are also no wide items remaining in T', apply the methods of Case 1. Otherwise we use the following process to free some space in the bins for wide and small items, i.e., $B_2, \ldots, B_\ell$. The idea of the process is to move small items from bins B_i to bins C_i and thereby move the tiny high items $T' \cap H$ further in direction C_ℓ. To do this, let $S_i = L_i \cup S$ be the set of small items in B_i.

Remove the tiny items from $C_2, \ldots, C_\ell$. If there exists an item $r \in S_i \cap B_i$ for some $i \in \{2, \ldots, \ell\}$ then remove r from B_i and add it to C_i, otherwise stop. Adding r to C_i is possible as C_i is a subset of $L_i = I_i^*$ and thus feasible. Add wide items from $W \cap T'$ to B_i until $\mathcal{A}(B_i) \geq 1/2 - \varepsilon$ again or $W \cap T' = \emptyset$. Finally, add the high items from $H \cap T'$ to $C_2, \ldots, C_\ell$ in a greedy manner analogously to Step 4 of the first part of the algorithm but using the area bound $\mathcal{A}(C_i) \leq 1/2$. This ensures that all sets C_i can be packed with Steinberg's algorithm. Repeat this process until $S_i \cap B_i = \emptyset$ for all $i \in \{2, \ldots, \ell\}$ or T' contains a high item at the end of an iteration.

There are two ways in which this process can stop. First if we moved all items from S_i to C_i, and second if in the next step a high item would remain in T' after the process. In the first case we have reached a situation as in Case 2 or Case 3, i.e., the roles of the wide and the high items are interchanged and $\mathcal{A}(B_\ell) \geq 1/2 - \varepsilon$. Thus by rotating all items and the packing derived so far, we can solve this case analogously to Case 2 or Case 3, depending on $\mathcal{A}(C_\ell)$.

In the second case, let r^* be the item that stopped the process, i.e., if r^* is moved from B_i to C_i for some $i \in \{2, \ldots, \ell\}$, at least one high item would remain in T'. Then, instead of moving r^* to C_i we move r^* to C_1 and add items from T' to C_1 and C_i as long as $\mathcal{A}(C_1) \leq 1/2$ and $\mathcal{A}(C_i) \leq 1/2$. The resulting sets can be packed with Steinberg's algorithm as no item has width greater than $1/2$. If after this step still items remain unpacked then a calculation similar to Case 1 gives a total packed area of

$$\begin{aligned} A &\geq \mathcal{A}(B_1) + \overbrace{(2\ell - 2)\left(\frac{1}{2} - \varepsilon\right)}^{\text{all bins except } B_1, B_i} + \overbrace{\frac{1}{2} - \varepsilon - \mathcal{A}(r^*)}^{\text{bin } B_i} \\ &\geq \ell\mathcal{A}(I_1^*) + \frac{1}{2} - 2\ell\varepsilon - \mathcal{A}(r^*) > \ell\mathcal{A}(I_1^*) \qquad \text{since } \mathcal{A}(r^*) \leq 1/4 \text{ and } \varepsilon < 1/(8\ell). \end{aligned}$$

From Inequality (2) we have the contradiction $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$. Thus all items are packed.

Note that we use crossreferences between the four cases but there are no circles in these references, i.e., Case 3 uses Case 1, and Case 4 uses Cases 1, 2 and 3.

We showed the following lemma.

Lemma 9. *There exists a polynomial-time algorithm that, given an instances I with $1 < \mathrm{OPT}(I) < k$, returns a packing in $2\mathrm{OPT}(I)$ bins.*

6 The overall algorithm

Let us recapitulate the different cases of our algorithm. We use the asymptotic algorithm by Bansal et al.[2] that solves instances I with $\mathrm{OPT}(I) \geq k$ for some constant k. As we do not know the optimal value in advance, we apply our algorithms in any case but each algorithm is allowed to fail if its requirement on $\mathrm{OPT}(I)$ is not satisfied.

For $\mathrm{OPT}(I) = 1$ we presented an algorithm in Section 4 that returns a packing into two bins. The algorithm is based on two major cases according to the total height of the δ-wide and the total width of the δ-high items. Either these height (width) is suitably bounded (for some $\delta \in (\varepsilon, 1/2]$) in which case we apply the methods of Lemma 2, or we get a substantial area guarantee. We utilize this area guarantee in the methods of Lemmas 6 and 8.

For $1 < \mathrm{OPT}(I) < k$ we presented an algorithm in Section 5 that is based on an enumeration of some large items and a separation of the wide and the high items. Lemma 9 shows that the algorithm outputs a packing into at most $2\mathrm{OPT}(I)$ bins. In total we showed the following theorem.

Theorem 3. *There exists a polynomial-time 2-approximation algorithm for two-dimensional bin packing.*

We cannot give an explicit running time of our algorithm as it is based on the BCS algorithm for which the running time is only stated as polynomial in n for any fixed $\varepsilon > 0$ and $r > 1$. The running time of our algorithm is also bounded by some polynomial in n.

Also note that in order to implement our algorithm one needs to know the constant k. It is not made explicit in in paper of Bansal et al.[2] but can in principle be bounded from above by their methods.

References

1. N. Bansal. personal communication, 2008.
2. N. Bansal, A. Caprara, and M. Sviridenko. Improved approximation algorithms for multidimensional bin packing problems. In *FOCS: Proc. 47nd IEEE Symposium on Foundations of Computer Science*, pages 697–708, 2006.
3. N. Bansal, A. Caprara, and M. Sviridenko. A structural lemma in 2-dimensional packing, and its implications on approximability, 2008. IBM Research Division, RC24468 (W0801-070), http://domino.research.ibm.com/library/cyberdig.nsf/index.html.
4. N. Bansal, J. R. Correa, C. Kenyon, and M. Sviridenko. Bin packing in multiple dimensions - inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31(1):31–49, 2006.
5. A. Caprara. Packing 2-dimensional bins in harmony. In *FOCS: Proc. 43rd IEEE Symposium on Foundations of Computer Science*, pages 490–499, 2002.
6. A. Caprara, A. Lodi, and M. Monaci. Fast approximation schemes for two-stage, two-dimensional bin packing. *Mathematics of Operations Research*, 30(1):150–172, 2005.
7. M. Chlebík and J. Chlebíková. Inapproximability results for orthogonal rectangle packing problems with rotations. In *CIAC: Proc. 6th Conference on Algorithms and Complexity*, pages 199–210, 2006.
8. R. Harren and R. van Stee. Absolute approximation ratios for packing rectangles into bins. *Journal of Scheduling*, accepted for publication.
9. K. Jansen and R. Thöle. Approximation algorithms for scheduling parallel jobs: Breaking the approximation ratio of 2. In *ICALP: Proc. 35rd International Colloquium on Automata, Languages and Programming*, pages 234–245, 2008.
10. K. Jansen and G. Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007.
11. J. Y.-T. Leung, T. W. Tam, C. S. Wong, G. H. Young, and F. Y. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275, 1990.
12. I. Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *ESA: Proc. 2nd European Symposium on Algorithms*, pages 290–299, 1994.
13. A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, 26(2):401–409, 1997.
14. R. van Stee. An approximation algorithm for square packing. *Operations Research Letters*, 32(6):535–539, 2004.
15. G. Zhang. A 3-approximation algorithm for two-dimensional bin packing. *Operations Research Letters*, 33(2):121–126, 2005.